ABSTRACT

The authors describe a methodology for evaluating the performability (combined performance and reliability) of large communications networks. Networks are represented by a 4-level hierarchical model, consisting of traffic matrix, network graph, “components” representing failure modes, and reliability information. Network states are assignments of modes to the network components, which usually represent network elements and their key modules, although they can be more abstract. The components can be binary or multi-modal, and each of their failure modes may change a set of attributes of the graph (e.g. the capacity or cost of a link). Their methodology also captures the effect of automatic restoration against network failures by including two common rerouting methods. To compute network performability measures, including upper and lower bounds on their cumulative distribution functions, we augment existing probabilistic state-space generation algorithms with our new “hybrid” algorithm. To characterize the network failures of highest impact, we compute the Pareto boundaries of the network’s risk space. The authors have developed a network analysis tool called nperf that embodies this methodology. To illustrate the methodology and the practicality of the tool, they describe a performability analysis of three design alternatives for a large commercial IP backbone network.

INTRODUCTION

Network service providers usually guarantee a certain level of service to their enterprise customers by specifying limits on network measurements such as down time, restoration delay, and packet delay. An example of such a guarantee,
also called a “service-level agreement” (SLA), is “with 99.99% probability, at most 5% of the network traffic will be unavailable”. Violation of these agreements results in a penalty to the service providers. In turn, the enterprise customers incur business losses if their networks do not perform as expected. Some aspects of the service level involve only reliability, such as downtime, while others are pure performance measures, such as packet delay. Each can be evaluated or predicted in isolation by an appropriate model and technique. However, as networks become larger and more complex, it is less realistic to consider only pure reliability measures or to evaluate performance measures as if the network were always in its perfect (no-failure) state.

Even though the complexity of real networks often requires that performance and reliability evaluations be treated separately from each other, a combined, simultaneous evaluation, known as performability analysis, is more useful (Meyer, 1995). Performability analysis characterizes failures probabilistically and evaluates a performance measure (or measures) over a large set of network states, one of which is the no-failure state and the rest represent combinations of failures. Then one can estimate the expected value of a measure or estimate the probability that the measure does not exceed a threshold. These results often expose the inadequacy of separate performance and reliability analyses, but the simultaneous analysis is much more difficult. For example, if the effects of restoration after network failure are included, performability computation becomes significantly more complex. See Colbourn (1999) for some more on network reliability vs. performability evaluation.

Network performability, as well as stand-alone reliability or performance models, fall into two basic categories. One approach is to consider the effect of equipment failure and network restoration on the performance of a fixed “reference” connection, chosen to represent either the typical or the worst case. This allows very detailed and precise results to be obtained (Oikonomou, 2006), but ignores the interaction among connections in a network. The other approach, which we employ here, is a performability analysis of the entire network. There are two main points to note. First, the complexity of any realistic network necessitates a description in various layers and a failure mapping between these layers. Second, because the size of the state space is exponential in the number of failure modes, exploring all possible failure states is impractical. An approach proven successful in practice is to generate failures in order of probability, i.e. most likely first, and evaluate the performance measures on these states as they are generated.

Related Work

Colbourn (1999); Harms, Kraetzl, Colbourn, and Devitt (1999); and Sahner, Trivedi, and Puliafito (1996) provide background on reliability and performability evaluation. Algorithms for generating the states of a system in order of probability are in Yang and Kubat (1990) (1989); Gomes and Craveirinha (1998); and Jarvis and Shier (1996). Network performability studies have also appeared in the literature. Levy and Wirth (1989) study response time in a multiprocessor system and call completion rate in a communications network. Alvarez, Uysal, and Merchant (2001) study performability guarantees for the time required to satisfy a web request in a network with up to 50 nodes, where only nodes can fail, and without restoration. Levendovszky, Jereb, Elek, and Vesztergombi (2002) study the expected lost traffic in the Hungarian backbone SDH network with 52 nodes and 59 links, and no restoration. Carlier, Li, and Lutton (1997) use a 3-level network model, and study expected lost traffic in a 111-node, 180-link network using k-shortest path restoration. Gomes and Craveirinha (1997) study a 46-node, 693-link representation of the Lisbon urban network with a 3-level performability model,